International Journal of Computer Science and Engineering (IJCSE)
Vol.1, Issue 1 Aug 2012 15-24
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations
IASET

DEVELOPMENT OF AN EFFICIENT FUZZY INTEGRATED QUALITY FUNCTION DEPLOYMENT SOFTWARE - A CONCEPTUAL ANALYSIS

JITENDRA SHARMA

Associate Professor, Institute of Management Technology, Nagpur, India

ABSTRACT

Quality Function Deployment (QFD) is a methodology for building the "Voice of the Customer" into product and service design. It is a tool which captures customer requirements and translates those needs into characteristics about a product or service. In the Quality Function Deployment (QFD) process, decision making is an essential and crucial task. QFD is an extensive process that contains loads of data and involves complex calculations making it more tedious for designers and engineers to deal manually with this data. Moreover, since the traditional QFD exercise encounters some problems like - use of linguistic expressions and crisp values, fuzzy concepts are to be employed for better results. Thus a need for efficient fuzzy integrated QFD software is highly recognized in the QFD software market. Softwares can be suitably designed to meet market requirements only when the associated data are meticulously examined and customer needs are better understood. To this end, the paper aims to analyze the QFD process from both viewpoints – Traditional as well as Software so as to mine valuable information which can be used for the development of QFD software. It then talks about the shortcomings in the available ones and the features required. Finally it concludes with the discussion of fuzzy concepts and its incorporation in the QFD software. The result of this work will assist the software developers in understanding the QFD process and choosing the appropriate tools for development of QFD software.

KEY WORDS: Automation, Data, Fuzzy, QFD, TQM, Software.

INTRODUCTION

QFD is a total quality management (TQM) tool ubiquitously used for quality product development. Yoji Akao defined QFD as “a method for developing a design quality aimed at satisfying the consumer and then translating the consumer's demands into design targets and major quality assurance points to be used throughout the production phase" [1]. One typical misunderstanding of many QFD users is the reduction of the QFD methodology [2] to the House of Quality [3]. QFD means more than the House of Quality. There are many other tools (e. g. Voice of the Customer Table, affinity- and tree-diagrams, comparisons in pairs, importance satisfaction-portfolios, Pareto-analyses, etc.) for analyzing customer needs and satisfaction, gathering and examining market information, etc [4]. Quality Function Deployment (QFD), identified as the implementation vehicle of Total Quality Management (TQM), has been proposed as an effective approach for implementing quality improvement programs.

Although several QFD softwares are available in the market today, but either they lack some required features or are not user friendly. The commercially available QFD software packages do little more than manipulating a House of Quality. Further, their instruction manuals show QFD as it was practiced more than 20 years ago [5]. Also, the conventional QFD exercise has some problems like - use of linguistic expressions and crisp values which yields inconsistency in the results. The issue can be better addressed using fuzzy QFD. As a result, there is a pressing need for user-friendly and efficient QFD software.

In this regard, the paper, analyzes QFD from both viewpoints – Traditional as well as Software. In the software viewpoint section, need of QFD software, shortcomings in the available ones, desired features in the QFD software are discussed. The section also discusses fuzzy concepts and implementation in the QFD software. Software analysis of QFD basically focuses on highlighting the nature and important modules of QFD so as to mine vital details that could assist software developers in developing efficient QFD software. The result of this work will greatly assist the software developers in understanding the QFD process and choosing the appropriate tools for development of QFD software.

QUALITY FUNCTION DEPLOYMENT (QFD): TRADITIONAL VIEWPOINT

QFD was first introduced in Japan by Akao in 1966 [6]. Quality function deployment is a system to assure that customer needs drive the product design and production process. It is a TQM technique that provides a means of translating customer requirements into the appropriate technical requirements for each stage of product development. Quality Function Deployment (QFD) is one of the key quality systems of TQM which was built to assure value to customers [7]. The true meaning of the phrase QFD is customer driven product (or process or service) development. It is a system for translating customer requirements into appropriate company requirements at each stage, from research and product development, to engineering and manufacturing, to marketing/sales and distribution. QFD is a disciplined, systematic method that ensures the voice of the customer is heard throughout the development, manufacturing, and product launch processes. Some say that QFD is a way to "neutralize the voice of the executives or engineers" [8].

QFD is based on the widely used Four Phase Model which focuses on the quality deployment part of Akao’s comprehensive QFD framework. The first matrix corresponds to the House of Quality (HOQ), which transforms customer requirements into measurable quality elements. The most important of these quality elements are then set against the characteristics of possible product components in a second matrix. These in turn correlate with the central process parameters in the third matrix, which are connected with definite production plans and means of production in the fourth matrix. So the four phases represent product planning, component planning, process planning and production planning. QFD is an innovative approach bringing quality as demanded by the customers – upstream into the product development process. Quality Function Deployment is a systematic process for helping a business to focus on its priorities, investments and its customers. Yoji Akao is widely regarded as the father of QFD and his work led to its first implementation at the Mitsubishi Heavy Industries Kobe Shipyard in 1972.

The QFD comprises of several different rooms which are sequentially filled in order to achieve an actionable translation from requirements into characteristics. HOQ is formed by closely linked building blocks.

The major advantage of the QFD process is that it encourages proactive product development instead of reactive product development (Figure 1). Proactive product development results in fewer and earlier design changes, decreased development time, fewer start-up problems, lower start-up costs, fewer field problems, and a more satisfied customer. A less obvious, though equally important, benefit of QFD is that it facilitates organizational knowledge transfer and establishes a proprietary knowledge base. The matrices that are generated during a QFD project make the logic flow obvious and act to preserve technical and customer knowledge. This affords others in the organization the opportunity to easily access and use the accumulated knowledge [8].

THE HOUSE OF QUALITY

The "House of Quality" (HOQ) matrix is the most recognized form of QFD. It is utilized by a multidisciplinary team to translate a set of customer requirements into an appropriate number of prioritized engineering targets to be met by a new product design. HOQ is the matrix which analyses customer requirements in detail and translates them into the developer’s language [9]. There are many slightly different forms of this matrix and this ability to be adapted to the requirements of a particular problem or group of users forms one of its major strengths. Quality Function Deployment uses the House of Quality (HOQ) as a visual model (Refer Fig. 1). Generally speaking, the HOQ is the matrix which analyzes customer requirements in detail and translates them into the developers’ language. The HOQ is the framework of most of the matrices used in QFD.

In QFD approach, the matrix to be built is the Product Planning Matrix, also called House of Quality due to its house-like shape. Its purpose is to translate important customer requirements regarding product quality into key end-product control characteristics. The HoQ is the framework of most of the matrices used in QFD. Quality Function Deployment uses the House of Quality (HoQ) as a visual model (Fig. 1). The central tool of the Quality Function Deployment is the matrix chart called House of Quality. This tool is a powerful way of generating specific, prioritized, and measurable technical requirements from often ambiguous customer needs. It generally consists of six different areas, also called by the generic names of WHAT, HOW, WHAT-HOW, WHY, HOW-HOW, and HOW MUCH. The basic idea is to set certain requirements that are given (WHAT) against (WHAT-HOW) possible technical solutions (HOW). To make the house complete, concrete information about the existence of the requirements (WHY), the existing correlations among the possible technical solutions (HOW-HOW) and detailed development targets (HOW MUCH) are added. However, these generic names don’t fit the original content of the HoQ, because customer requirements rather represent the reasons *why* something is demanded and the product characteristics indicate *what* exactly is demanded. Creating the HoQ is often mistaken to be the same as QFD, it is only one matrix among several, although the most important one. Despite this, applying QFD always leads to the creation of the HoQ to form the basis for all further activities. It is also for this reason that the HoQ represents the heart of any QFD application in product planning.

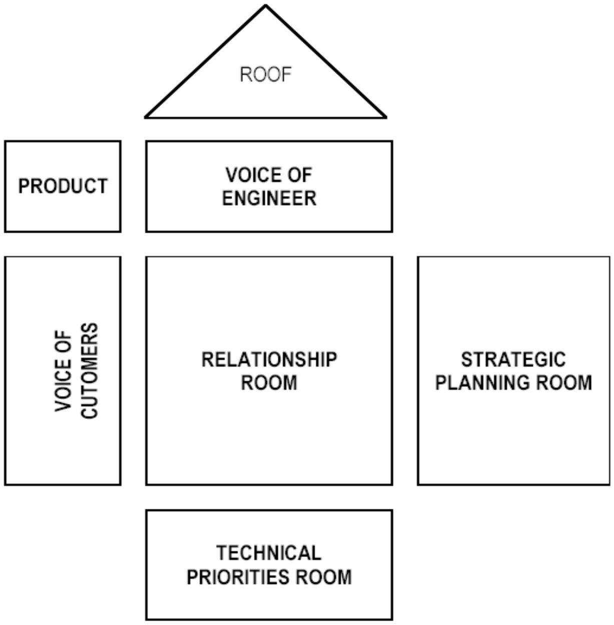

Figure 1: Basic Building Blocks of QFD - House Of Quality

QFD - SOFTWARE VIEWPOINT

QFD Software: Need and Shortcomings

QFD is very powerful as it incorporates the voice of the customer in the designs - hence it is likely that the final product will be better designed to satisfy the customer's needs. QFD is applied in the early stages of the design phase so that the customer wants are incorporated into the final product. In companies, the designers and engineers often have to deal with evaluations in decision making process with respect to product design and development. However these things get very complicated and tough due to the nature of traditional QFD exercise.

QFD makes use of a set of matrices. The process is a lengthy one which involves large amount of data and complex calculations. As a result, dealing with the data manually is quite cumbersome and time consuming. The issue can be better handle with the use of suitable software. However, the production of software calls for techniques such as specification, design, implementation, testing, and maintenance. Essential to performing the last three phases of software development is the selection of a programming language that acts as an implementation vehicle. Creating a software system using a software engineering process contains three main tasks or phases: the functions and features of the expected software have to be defined, the software has to be implemented and it has to be deployed in an operating environment. The functions and features of the expected software are called requirements [10].

As a result, the companies involved in QFD are continuously seeking for efficient QFD software that could assist designers and engineers to evaluate company's decision making process with respect to product design and development. Software is an intangible product that is not always conducive to explicit acceptance measures. Design elements are coupled and interdependent, which is different from physical designs that can be deconstructed into independent but functional sub-assemblies, parts and components [11].

Commercial QFD softwares currently available in the market suffer from the limitation to simply draw matrices. Moreover, some applications even fail to address the House of Quality tool entirely. Some even rely on spreadsheet technology for building their Houses of Quality (HOQs). The QFD-software should provide not only means of entering data into matrices and print them out, but also helps to generate, compute and analyze the data. Moreover, conventional QFD exercise has some problems like - use of linguistic expressions and crisp values which yields inconsistency in the results. The issue can be resolved using fuzzy QFD.

Desired Features of a QFD Software

QFD software should aim to create and maintain Quality Function Deployment matrices. Suitable QFD software can not only assist a company in decision making process but also helps in executing the process with more accuracy and speed. Moreover, In general, a good software program is meant for things that are difficult, time-consuming, or impossible to do otherwise. Software can be defined as programs, procedures, rules and any associated documentation pertaining to the operation of a computer system [12].

1. It should function beyond drawing matrices and tables.
2. It should be user friendly and efficient
3. It should be fast and simple in usage to prepare tables and matrices.
4. It should primarily aim at the following primary characteristics: functionality, reliability, usability, efficiency and maintainability.
5. It should help in planning products, services or strategies with fewer corrections.
6. It should provide help facility
7. It should support strong interaction with other applications
8. It should support import and export operations.
9. It should offer great ease of operability
10. It should be of great help to engineers and designers in the decision making process with respect to product design and development.

QFD - Fuzzy Aspects

QFD is an exercise that has some problems like - use of linguistic expressions and crisp values which yields inconsistency in the results. Although the quality of a product can be dramatically improved through a QFD exercise, the traditional crisp scoring approach has a major drawback. A wrong conclusion can be easily produced since the fuzzy nature of linguistic correlation terms from evaluation members is ignored. To overcome this problem, fuzzy scoring for linguistic terms can be utilized.

Fuzzy logic and fuzzy sets were introduced in 1965 by Professor L.A. Zadeh. Fuzzy logic uses human linguistics (word or sentences) understanding to express the knowledge of a system. Fuzzy set theory has been proven as a useful tool in modeling the intuition, vagueness, and imprecision presented in descriptions of a decision-making or optimization problem [15]. Fuzzy analysis is capable of dealing with qualitative or imprecise inputs from designers by describing the performance of each criterion with some linguistic terms, such as “good”, “poor”, “medium”, *etc.* These linguistic terms can be represented and manipulated based on fuzzy set theory. Symmetrical Triangular Fuzzy Numbers (STFNs) can be employed to capture the vagueness. Use of fuzzy concepts at different sections of QFD has already been reported in various articles. A scope for fuzzy concepts to be implemented broadly exists in QFD. Presence of fuzzy implementations in the QFD Software will greatly benefit the process in bringing more accurate results. This knowledge consists of facts, concepts, theories, procedures, and relationships and is expressed in the form of IF-THEN rules. Linguistic variables are characterized by ambiguity and multiplicity of meaning. Specifying good linguistic variables depends on the knowledge of the expert. For example “age” is a linguistic variable if its values are “young”, “not so young”, “old”, and “very old”. In fuzzy logic theory, a linguistic variable can be a member of more than one group.

Fuzzy logic exhibits some useful features for exploitation in QFD. These include:

1. It uses human linguistic to express the knowledge of the system
2. It allows decision making with estimated values under incomplete or uncertain information.
3. It is suitable for uncertain or appropriate reasoning.
4. Interpretation of its rules is simple and easy to understand.
5. It deals with multi input and multi output system.

Various inputs, in the form of judgments and evaluations are needed in the QFD charts. Normally, these inputs are gathered through questionnaires, deep interviews, and focus groups. This gives rise to uncertainties when trying to quantify the information. Fuzzy logic can be used, in order to reduce the uncertainty of the collected data [16].

QFD - Software Analysis

A software development process can also be defined as a set of activities needed to transform the user requirements into a software system [13]. Software automation of an application has never been simplistic. Moreover the task gets more difficult if the business to automate happens to be too lengthy as it may involve the presence of massive data, numerous calculations and more time-consuming processes. Automation of such task enforces meticulous examination of the processes involved, gathering of data and then transcripting these data into the technical slabs for effective utilization. Besides, it also concerns with the tools selected for the automation purpose. The software development of an application involves looking into various aspects and is a very detailed process. Development of a software application can be accomplished using different programming tools however what carries importance is not the completion of task but the flexibility and efficiency that it offers. However, the methodology comprises of understanding the data flow of QFD and then highlighting the main components of the QFD. The important phase in any software development process is analyzing the business meticulously. The analysis comprises of studying the business from traditional as well as software development point of view. QFD uses a series of matrices to document information collected and developed and represent the team's plan for a product. The QFD methodology is based on a systems engineering approach consisting of the following general steps:

1. Obtain product requirements or technical characteristics from customer needs
2. Build product concepts to satisfy these requirements.
3. Evaluate product concepts to select most optimum (Concept Selection Matrix).
4. Partition system concept or architecture into subsystems or assemblies and flow-down higher-level requirements or technical characteristics to these subsystems or assemblies.
5. Derive lower-level product requirements (assembly or part characteristics) and specifications from subsystem/assembly requirements (Assembly/Part Deployment Matrix).
6. For critical assemblies or parts, flow-down lower-level product requirements (assembly or part characteristics) to process planning.
7. Determine manufacturing process steps to meet these assembly or part characteristics.
8. Based in these process steps, determine set-up requirements, process controls and quality controls to assure achievement of these critical assembly or part characteristics [14].

QFD is an exercise that comprises of set of matrices, is a lengthy process, involves understanding many concepts, dealing with massive data and complex calculations. QFD also includes presenting data in hierarchical tree format and representing the data in graphical form. QFD is based on the widely used Four Phase Model which focuses on the quality deployment part of Akao’s comprehensive QFD framework. The four phases represent product planning, component planning, process planning and production planning. HoQ is the heart of any QFD application. In addition to the "House of Quality" matrix, QFD utilizes several management and planning tools and some main processes which are used in many of its procedures: . A number of techniques have to be combined in order to get all information that is necessary to form the matrices and to exhaust the potential of QFD as far as possible.

1. Voice of Customer
2. Brainstorming
3. Affinity diagrams.
4. Benchmarking
5. Hierarchy trees.
6. Matrices and tables

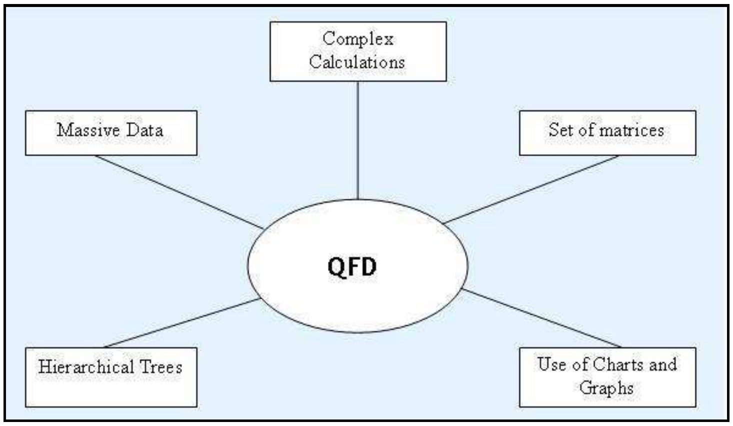

Figure 2: Parameters Involved in QFD

Fig. 2 explicates the nature of QFD. Automation of abovementioned modules will involve presenting data in hierarchical tree format, representing the data in graphical form and dealing with external operations. In voice of customer section, the data may come from several sources so the software can support import and export data operations. Brainstorming relates to extracting the useful data from the collection and then organizing these data through the use of affinity process. Benchmarking can be considered as making competitive analyses and involves greater computations. The task can be accomplished through the use of suitable tools and thus require to look for suitable ones.

Moreover, there are many scales used in HOQ process to measure different concepts. The linguistic terms people use to express their judgments are vague in nature. Using precise numbers to represent linguistic assessments are although used but are not very reasonable. Thus fuzzy numbers can be assigned to linguistic assessments to capture the vagueness. For example, instead of using numbers 1 and 9 to represent “very weak” and “very strong”, Symmetrical triangular fuzzy numbers (STFNs) can be better assigned as [0.5,1.5] and [8.5, 9.5] for these two linguistic assessments. Here an STFN, in the form of [a, c], is a special fuzzy set representing a fuzzy concept “approximately b” where b = (a + c) / 2 [17].

Analysis of QFD reveals that the QFD process can be better automated with the general purpose programming language. Besides the general features required in any language, some functionality like – displaying data in hierarchical tree structure, displaying charts and graphs, drag and drop operations, strong database interaction etc., are required to be implemented in QFD. Fuzzy concepts can be incorporated in QFD software to remove the vagueness and generate the values with more accuracy. Symmetrical Triangular Fuzzy Numbers (STFNs) can be employed to capture the vagueness.

CONCLUSIONS

Albeit several QFD softwares are available in the market but still the dearth of suitable QFD software terribly exists. Moreover, to capture the vagueness in people's linguistics assessments, incorporation of fuzzy concepts in the QFD software is highly sought for. In this regard, here, QFD is analyzed from traditional as well as software viewpoints. Through the presented research, an attempt has been made to bring out every important fact by analyzing the QFD process meticulously in order to help QFD software developers. In the traditional study viewpoint, conventional QFD process has been discussed while in the software viewpoint study, QFD has been analyzed from the software development perspective. Important modules in QFD and their nature of functioning have been discussed and possible automation of these has been presented in the software analysis section. This section also briefs the suitability of programming language in this regard. Further fuzzy concepts in QFD and their (STFNs) implementation have been discussed. Once the nature of QFD is understood then the selection of appropriate tools can be efficiently done. Thus, it has been realized that the research work presented here will certainly help in knowing the QFD process, selecting the appropriate tools and thus developing the efficient fuzzy integrated QFD software for better decision making and product development.

REFERENCES

1. Akao,Yoji .Quality function deployment : Integrating customer requirements into products design . Cambridge, Massachusetts, 1990.

2. General works on QFD: Yoji Akao: Quality Function Deployment: Integrating Customer Requirements into Product Design. Translated by Glenn H. Mazur and Japan Business Consultants, Ldt. Cambridge, Massachusetts 1990; Bob King: QFD. Better Designs in Half the Time. Methuen, Massachusetts 1987; especially in the field of software production: Richard E. Zultner: Quality Function Deployment (QFD): Structured Requirements Exploration. In: G. Gordon Schulmeyer, James I. McManus (eds.): Total Quality Management for Software. New York - London 1992, pp. 297-319.

3. Georg Herzwurm, Sixten Schockert, and Werner Mellis: Determining the Success of a QFD project - exemplified by a pilot scheme carried out in cooperation with the German software company SAP AG. In: QFD Institute (Ed.): Proceedings of the Eighth Symposium on Quality Function Deployment and International Symposium on QFD '96, June 9-11, 1996 in Novi, Michigan, USA. Novi 1996, pp. 131-150.

4. Georg Herzwurm, Sixten Schockert, Werner Mellis: Qualitätssoftware durch Kundenorientierung.

5. Die Methode Quality Function Deplyoment (QFD). Grundlagen, Praxisleitfaden, SAP R/3 Fallbeispiel. Braunschweig - Wiesbaden 1997. http://www.mazur.net/qfd_software.htm.

6. George Herzwurm, sixten schockert, Werner mellis. Determining the success of QFD,QFD institute – Proceedings of the 8th Symposium on QFD Novi , Michigan, USA,1996.

7. Sharma, J.R. et al. Understanding quality function deployment-A TQM tool to quantify customers needs ‘. National conference on world class manufacturing, Coimbatore , 2003.

8. Quality Function Deployment (Qfd): A Case Study, Robin Rawlings-Quinn, Manager, Market Research And Developmental Process, Intertape Polymer Group, Marysville, Mi.

9. Abbie Griffin . Evaluating QFD‘s use in U.S. firms as a process for developing products, Journal of product innovation management, Michigan, USA,1992.

10. Applying QFD to improve the requirements and project management in small-scale project, Terhi Kivinen, University of Tampere, Department of Computing Sciences, Computer sciences.

11. Integrating Quality Function Deployment (Qfd) with software development engineering for higher customer satisfaction, J R. Sharma, Sangita Tabarno, Dr. A. M. Rawani.

12. Zultner, Richard E. Software Quality Function Deployment. Applying QFD to Software. Transactions from the Second Symposium on Quality Function Deployment.Novi, Michigan, 1990.

13. I. Jacobson, G. Booch, and I. Rumbaugh, The Unified Software Development Process. Addison-Wesley, Boston, 1998.

14. http://www.npd-solutions.com/qfdsteps.htm

15. Using Fuzzy QFD for Design of Low-end Digital Camera, Ming-Chyuan Lin a, Chieh-Yuan Tsai b, Chao-Chun Cheng c, and C. Alec Chang, International Journal of Applied Science and Engineering 2004. 2, 3: 222-233.

16. The Methodology of Quality Function Deployment with Crisp and Fuzzy Approaches and an Application in the Turkish Shampoo Industry, Selim Zaim, Mehmet Ṣevkli , Journal of Economic and Social Research 4 (1), 27-53.

17. A systematic approach to QFD with a full illustrative example, Lai-Kow Chan, Ming-Lu Wu, International journal of Management science, Omega 33 (2005) 119-139.